Seguridad en iOS y Android: un Análisis Comparativo

Tesina de Grado

Autor:
Raúl Ignacio Galuppo

Director:
Dr. Carlos Luna

marzo, 2018

Contenidos

1. Introducción
   - Motivación
   - Modelo de Android
   - Modelo de iOS

Contenidos

1. Introducción
   - Motivación
   - Modelo de Android
   - Modelo de iOS
2. Análisis Comparativo

Contenidos

1. Introducción
   - Motivación
   - Modelo de Android
   - Modelo de iOS
2. Análisis Comparativo
3. Hacia un Framework para la Comparación de Permisos

Contenidos

1. Introducción
   - Motivación
   - Modelo de Android
   - Modelo de iOS
2. Análisis Comparativo
3. Hacia un Framework para la Comparación de Permisos
4. Conclusiones y Trabajos Futuros

Motivación

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles,

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo las bancarias.

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo las bancarias.
- Actualmente, más del 99 % de los dispositivos móviles en el mercado tiene a iOS o Android como su sistema operativo.

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo las bancarias.
- Actualmente, más del 99 % de los dispositivos móviles en el mercado tiene a iOS o Android como su sistema operativo. El número actual de aplicaciones de Android en el mercado supera los 3.500.00 y para iOS asciende a más de 3.100.000.

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo las bancarias.
- Actualmente, más del 99 % de los dispositivos móviles en el mercado tiene a iOS o Android como su sistema operativo. El número actual de aplicaciones de Android en el mercado supera los 3.500.00 y para iOS asciende a más de 3.100.000.
- Además, debido al uso diario de estas aplicaciones, se puede filtrar una gran cantidad de información privada y confidencial

Motivación

- La seguridad en dispositivos móviles se ha convertido en un asunto muy importante debido al incremento de ataques recibidos y a las consecuencias que éstos tienen.
- Los ataques están incentivados por la popularización de los dispositivos móviles, el aumento de información personal y confidencial que almacenan y las operaciones realizadas a través de ellos, como por ejemplo las bancarias.
- Actualmente, más del 99 % de los dispositivos móviles en el mercado tiene a iOS o Android como su sistema operativo. El número actual de aplicaciones de Android en el mercado supera los 3.500.00 y para iOS asciende a más de 3.100.000.
- Además, debido al uso diario de estas aplicaciones, se puede filtrar una gran cantidad de información privada y confidencial a menos que se aplique control de acceso a las aplicaciones instaladas.

Motivación

Motivación I

Este trabajo realiza un análisis detallado sobre las características de seguridad en Android e iOS, con el objetivo de preservar la privacidad del usuario.

Motivación

Motivación I

Este trabajo realiza un análisis detallado sobre las características de seguridad en Android e iOS, con el objetivo de preservar la privacidad del usuario.

Motivación II

Sumado al análisis, se presenta un *framework* comparativo, cuyas principales funciones son:

- Determinar empíricamente los alcances de los sistemas de permisos.
- Establecer una relación entre los permisos presenten en ambas plataformas.

Modelo de Android

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.

[1] Traducción propuesta para el término *sandbox*.
[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.
- Su arquitectura sigue el estilo arquitectónico conocido como Sistemas Estratificados: los distintos componentes se agrupan en capas según su nivel de abstracción, conformando una jerarquía.

[1] Traducción propuesta para el término *sandbox*.

[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.
- Su arquitectura sigue el estilo arquitectónico conocido como Sistemas Estratificados: los distintos componentes se agrupan en capas según su nivel de abstracción, conformando una jerarquía. Las capas inferiores contienen componentes ligados al *hardware*, mientras que las capas superiores agrupan componentes ligados con tareas de más alto nivel.

[1] Traducción propuesta para el término *sandbox*.

[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.
- Su arquitectura sigue el estilo arquitectónico conocido como Sistemas Estratificados: los distintos componentes se agrupan en capas según su nivel de abstracción, conformando una jerarquía. Las capas inferiores contienen componentes ligados al *hardware*, mientras que las capas superiores agrupan componentes ligados con tareas de más alto nivel.
- Cada aplicación se ejecuta en un *entorno aislado*[1], forzando a que solo pueda tener acceso irrestricto a sus propios recursos.
- Los recursos que provee Android sólo pueden ser accedidos mediante una SS-API[2] con un doble objetivo:

[1] Traducción propuesta para el término *sandbox*.

[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.
- Su arquitectura sigue el estilo arquitectónico conocido como Sistemas Estratificados: los distintos componentes se agrupan en capas según su nivel de abstracción, conformando una jerarquía. Las capas inferiores contienen componentes ligados al *hardware*, mientras que las capas superiores agrupan componentes ligados con tareas de más alto nivel.
- Cada aplicación se ejecuta en un *entorno aislado*[1], forzando a que solo pueda tener acceso irrestricto a sus propios recursos.
- Los recursos que provee Android sólo pueden ser accedidos mediante una SS-API[2] con un doble objetivo: tenerlos aislados y permitir cierta granularidad de seguridad sobre ellos.

[1] Traducción propuesta para el término *sandbox*.

[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Android es un sistema operativo de código abierto, diseñado para dispositivos móviles y desarrollado por Google junto con la Open Handset Alliance.
- Su arquitectura sigue el estilo arquitectónico conocido como Sistemas Estratificados: los distintos componentes se agrupan en capas según su nivel de abstracción, conformando una jerarquía. Las capas inferiores contienen componentes ligados al *hardware*, mientras que las capas superiores agrupan componentes ligados con tareas de más alto nivel.
- Cada aplicación se ejecuta en un *entorno aislado*[1], forzando a que solo pueda tener acceso irrestricto a sus propios recursos.
- Los recursos que provee Android sólo pueden ser accedidos mediante una SS-API[2] con un doble objetivo: tenerlos aislados y permitir cierta granularidad de seguridad sobre ellos.
- El mecanismo para el acceso a estas SS-API se llama Permisos.

[1] Traducción propuesta para el término *sandbox*.

[2] *Security Sensitive API*, por sus siglas en inglés.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.
- En las versiones anteriores a Android Marshmallow, al prepararse para instalar una aplicación, el sistema operativo mostraba un diálogo al usuario indicando los permisos solicitados y se le consultaba si deseaba continuar con la instalación.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.
- En las versiones anteriores a Android Marshmallow, al prepararse para instalar una aplicación, el sistema operativo mostraba un diálogo al usuario indicando los permisos solicitados y se le consultaba si deseaba continuar con la instalación.
- En caso afirmativo, el sistema otorgaba todos los permisos solicitados e instalaba la aplicación.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.
- En las versiones anteriores a Android Marshmallow, al prepararse para instalar una aplicación, el sistema operativo mostraba un diálogo al usuario indicando los permisos solicitados y se le consultaba si deseaba continuar con la instalación.
- En caso afirmativo, el sistema otorgaba todos los permisos solicitados e instalaba la aplicación. En el caso contrario, no se instalaba la aplicación.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.
- En las versiones anteriores a Android Marshmallow, al prepararse para instalar una aplicación, el sistema operativo mostraba un diálogo al usuario indicando los permisos solicitados y se le consultaba si deseaba continuar con la instalación.
- En caso afirmativo, el sistema otorgaba todos los permisos solicitados e instalaba la aplicación. En el caso contrario, no se instalaba la aplicación.
- El usuario quedaba preso si quería instalar una aplicación: no podía otorgar o denegar permisos individuales; debía otorgar o denegar todos los permisos solicitados como un bloque.

Modelo de Android

- Podemos clasificar los permisos según el riesgo implícito al otorgarlos, resultando las siguientes cuatro categorías: *Normal*, *Dangerous*, *Signature* y *Signature/System*. El presente informe se centra en los permisos *Normales* y *Peligrosos*; cómo se otorgan y cómo se deniegan.
- En las versiones anteriores a Android Marshmallow, al prepararse para instalar una aplicación, el sistema operativo mostraba un diálogo al usuario indicando los permisos solicitados y se le consultaba si deseaba continuar con la instalación.
- En caso afirmativo, el sistema otorgaba todos los permisos solicitados e instalaba la aplicación. En el caso contrario, no se instalaba la aplicación.
- El usuario quedaba preso si quería instalar una aplicación: no podía otorgar o denegar permisos individuales; debía otorgar o denegar todos los permisos solicitados como un bloque. Una vez concedidos, los permisos seguían vigentes mientras la aplicación estuviera instalada. Solo se eliminaban si se desinstala dicha aplicación.

Modelo de Android

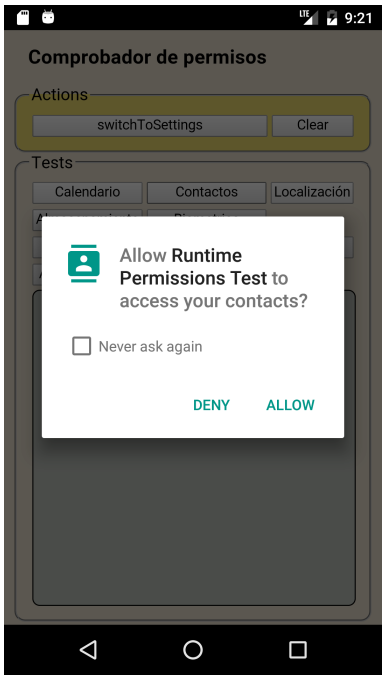
- A partir de la versión 6.0 se propone un nuevo modelo de permisos, donde los usuarios pueden administrar en tiempo de ejecución los permisos *peligrosos* requeridos por una aplicación.

Modelo de Android

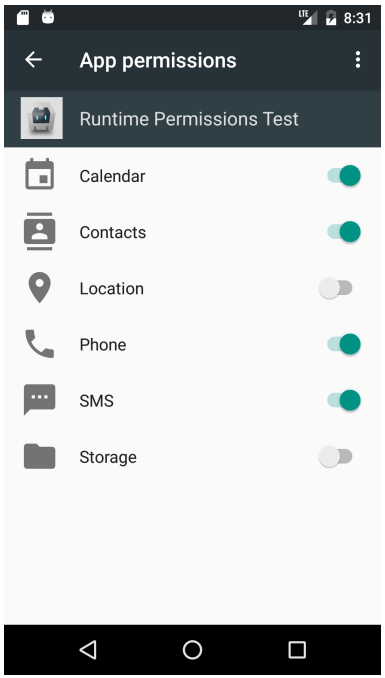
- A partir de la versión 6.0 se propone un nuevo modelo de permisos, donde los usuarios pueden administrar en tiempo de ejecución los permisos *peligrosos* requeridos por una aplicación. En este modelo, los permisos se agrupan para facilitar el control de la privacidad de los usuarios.

Modelo de Android

- A partir de la versión 6.0 se propone un nuevo modelo de permisos, donde los usuarios pueden administrar en tiempo de ejecución los permisos *peligrosos* requeridos por una aplicación. En este modelo, los permisos se agrupan para facilitar el control de la privacidad de los usuarios.



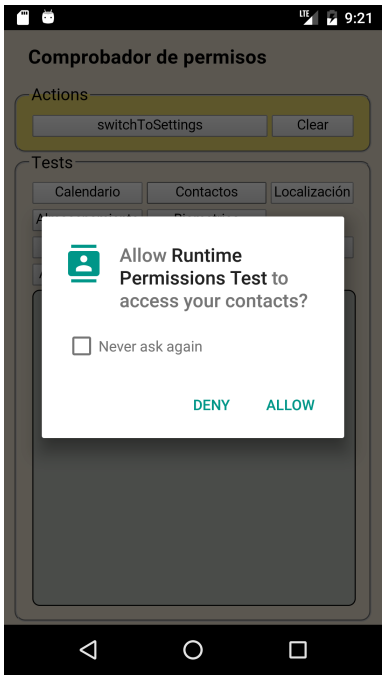
(a) Solicitud de un permiso.



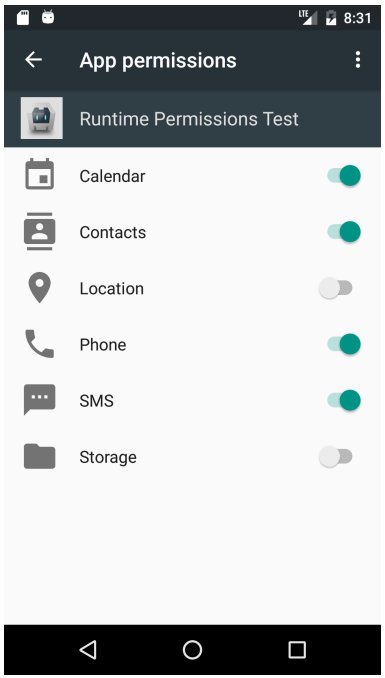
(b) Listado de los permisos.

Modelo de Android

- A partir de la versión 6.0 se propone un nuevo modelo de permisos, donde los usuarios pueden administrar en tiempo de ejecución los permisos *peligrosos* requeridos por una aplicación. En este modelo, los permisos se agrupan para facilitar el control de la privacidad de los usuarios.



(a) Solicitud de un permiso.

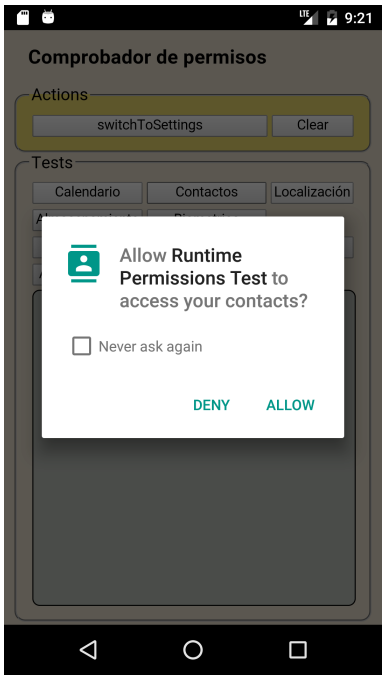


(b) Listado de los permisos.

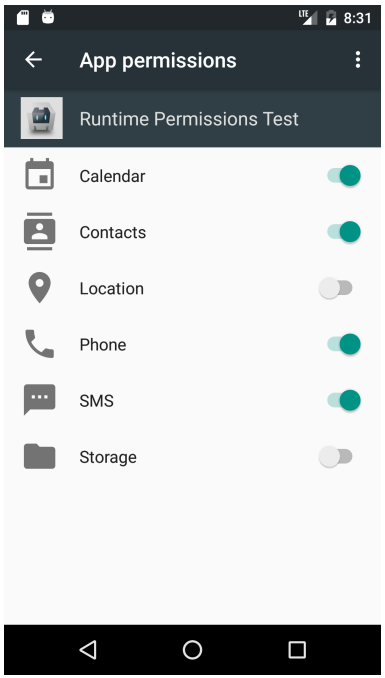
- A partir de esta versión, durante la instalación de una aplicación se le otorgan todos los permisos *normales* y ningún permiso *peligroso*.

Modelo de Android

- A partir de la versión 6.0 se propone un nuevo modelo de permisos, donde los usuarios pueden administrar en tiempo de ejecución los permisos *peligrosos* requeridos por una aplicación. En este modelo, los permisos se agrupan para facilitar el control de la privacidad de los usuarios.



(a) Solicitud de un permiso.



(b) Listado de los permisos.

- A partir de esta versión, durante la instalación de una aplicación se le otorgan todos los permisos *normales* y ningún permiso *peligroso*.
- Como consecuencia de esto, cada vez que una aplicación necesita acceder a un recurso protegido por un permiso *peligroso*, tiene que solicitarlo en tiempo de ejecución.

Modelo de iOS

Modelo de iOS

- iOS es un sistema operativo para dispositivos móviles de la multinacional Apple Inc. diseñado para ser seguro.

Modelo de iOS

- iOS es un sistema operativo para dispositivos móviles de la multinacional Apple Inc. diseñado para ser seguro.
- Cada dispositivo combina hardware, software y servicios, diseñados para trabajar conjuntamente para proveer seguridad y al mismo tiempo, que sea transparente para el usuario.

Modelo de iOS

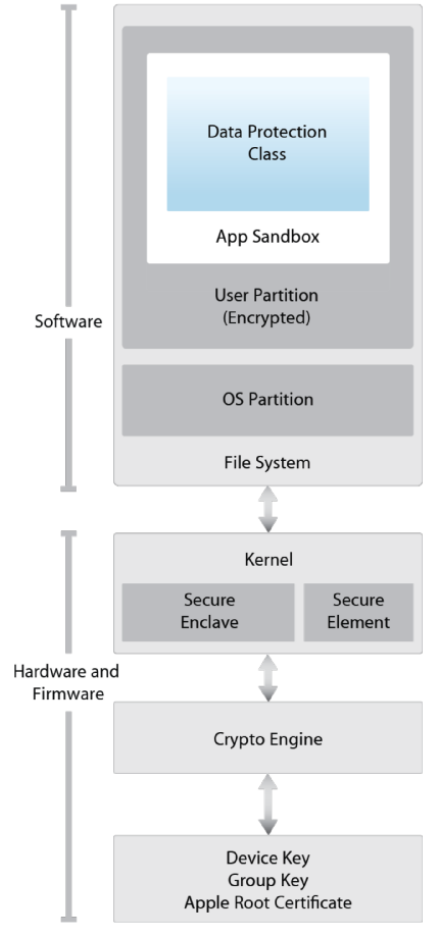
- iOS es un sistema operativo para dispositivos móviles de la multinacional Apple Inc. diseñado para ser seguro.
- Cada dispositivo combina hardware, software y servicios, diseñados para trabajar conjuntamente para proveer seguridad y al mismo tiempo, que sea transparente para el usuario.
- Las principales características de seguridad, como el cifrado del dispositivo, no son configurables y vienen habilitadas por defecto.

Modelo de iOS

- iOS es un sistema operativo para dispositivos móviles de la multinacional Apple Inc. diseñado para ser seguro.
- Cada dispositivo combina hardware, software y servicios, diseñados para trabajar conjuntamente para proveer seguridad y al mismo tiempo, que sea transparente para el usuario.
- Las principales características de seguridad, como el cifrado del dispositivo, no son configurables y vienen habilitadas por defecto.
- La seguridad se extiende más allá del dispositivo, incluido todo lo que hacen los usuarios localmente, en redes y con servicios clave de Internet, generando un ecosistema seguro.

Modelo de iOS

- iOS es un sistema operativo para dispositivos móviles de la multinacional Apple Inc. diseñado para ser seguro.
- Cada dispositivo combina hardware, software y servicios, diseñados para trabajar conjuntamente para proveer seguridad y al mismo tiempo, que sea transparente para el usuario.
- Las principales características de seguridad, como el cifrado del dispositivo, no son configurables y vienen habilitadas por defecto.
- La seguridad se extiende más allá del dispositivo, incluido todo lo que hacen los usuarios localmente, en redes y con servicios clave de Internet, generando un ecosistema seguro.



(a) Modelo de seguridad de iOS.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo. Si una aplicación requiere acceder a información que no es suya, lo puede hacer únicamente usando servicios de iOS.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo. Si una aplicación requiere acceder a información que no es suya, lo puede hacer únicamente usando servicios de iOS.
- iOS ayuda a evitar que las aplicaciones accedan a la información personal de un usuario sin permiso.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo. Si una aplicación requiere acceder a información que no es suya, lo puede hacer únicamente usando servicios de iOS.
- iOS ayuda a evitar que las aplicaciones accedan a la información personal de un usuario sin permiso.
- Para acceder a ciertos recursos necesita autorización explícita del usuario.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo. Si una aplicación requiere acceder a información que no es suya, lo puede hacer únicamente usando servicios de iOS.
- iOS ayuda a evitar que las aplicaciones accedan a la información personal de un usuario sin permiso.
- Para acceder a ciertos recursos necesita autorización explícita del usuario.
- Las aplicaciones pueden solicitar un permiso solamente mientras se esté ejecutando.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- Cada aplicación instalada por el usuario se ejecuta en un *entorno aislado*[3].
- Como consecuencia de esto, tiene denegado el acceso a los archivos guardados por otra aplicación y no puede realizar cambios en el dispositivo. Si una aplicación requiere acceder a información que no es suya, lo puede hacer únicamente usando servicios de iOS.
- iOS ayuda a evitar que las aplicaciones accedan a la información personal de un usuario sin permiso.
- Para acceder a ciertos recursos necesita autorización explícita del usuario.
- Las aplicaciones pueden solicitar un permiso solamente mientras se esté ejecutando. A su vez, los usuarios pueden optar por no permitir este acceso, y pueden cambiar su elección en cualquier momento.

[3] Traducción propuesta para el término *sandbox*.

Modelo de iOS

- De querer otorgar o revocar algún permiso, el usuario debe ir a la configuración de privacidad (`Ajustes/Privacidad`).

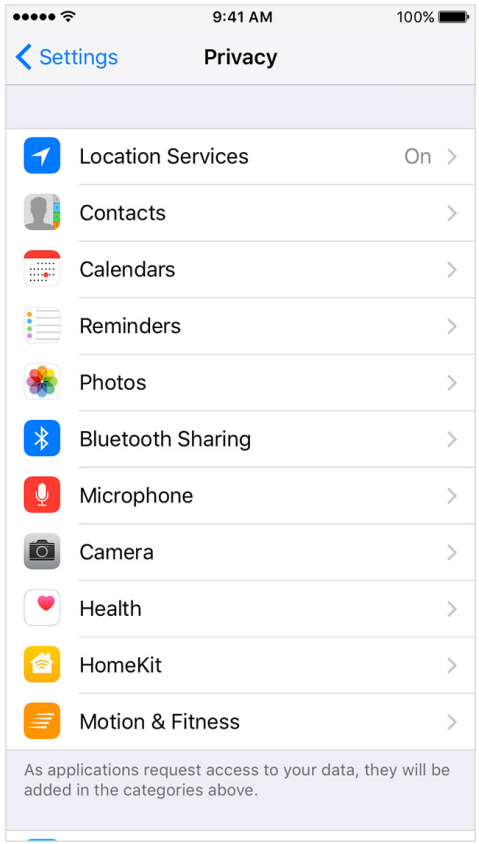


Figura : Control de privacidad de iOS 9.

Análisis Comparativo

Análisis Comparativo

- Existen muchos trabajos sobre distintas formas de comparar la seguridad de ambas plataformas.

Análisis Comparativo

- Existen muchos trabajos sobre distintas formas de comparar la seguridad de ambas plataformas.
- Cada uno de ellos propone una medida de comparación que focaliza el análisis en alguna característica particular de Android y/o de iOS.

Análisis Comparativo

- Existen muchos trabajos sobre distintas formas de comparar la seguridad de ambas plataformas.
- Cada uno de ellos propone una medida de comparación que focaliza el análisis en alguna característica particular de Android y/o de iOS.
- Esta tesina propone analizar distintas características presentes en ambas plataformas,

Análisis Comparativo

- Existen muchos trabajos sobre distintas formas de comparar la seguridad de ambas plataformas.
- Cada uno de ellos propone una medida de comparación que focaliza el análisis en alguna característica particular de Android y/o de iOS.
- Esta tesina propone analizar distintas características presentes en ambas plataformas,

Análisis Comparativo

- Existen muchos trabajos sobre distintas formas de comparar la seguridad de ambas plataformas.
- Cada uno de ellos propone una medida de comparación que focaliza el análisis en alguna característica particular de Android y/o de iOS.
- Esta tesina propone analizar distintas características presentes en ambas plataformas, poniendo foco en los permisos que se pueden modificar *en tiempo de ejecución*.

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado
- Cifrado del sistema de archivos

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado
- Cifrado del sistema de archivos
- Bloqueo del dispositivo

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado
- Cifrado del sistema de archivos
- Bloqueo del dispositivo
- Seguridad de las aplicaciones

Análisis Comparativo

Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado
- Cifrado del sistema de archivos
- Bloqueo del dispositivo
- Seguridad de las aplicaciones

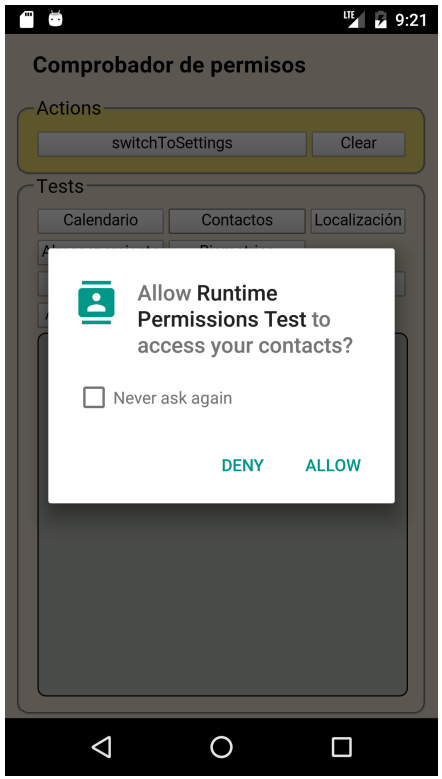
Análisis Comparativo

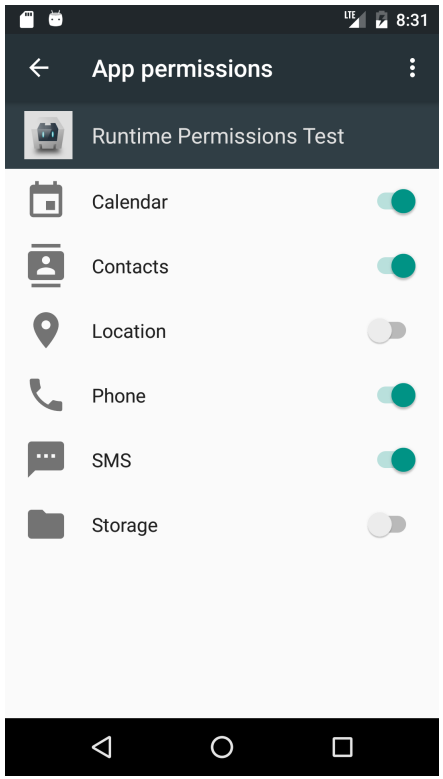
Se analizaron cuatro características presentes en iOS y Andriod:

- Arranque verificado
- Cifrado del sistema de archivos
- Bloqueo del dispositivo
- Seguridad de las aplicaciones

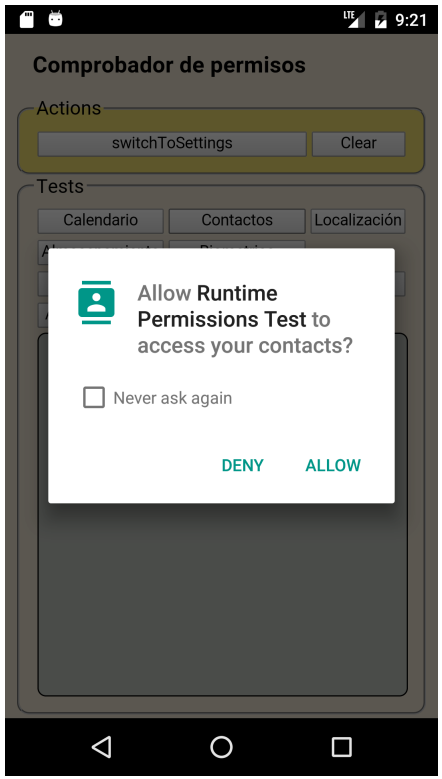
Se pone foco especialmente en los sistemas de permisos:

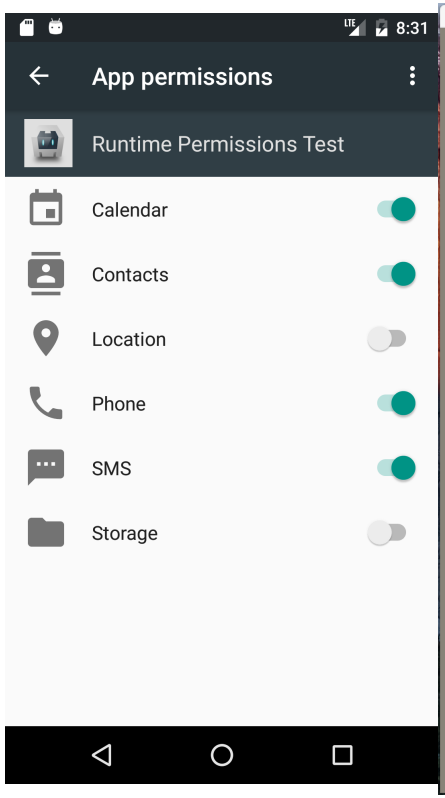
Análisis Comparativo

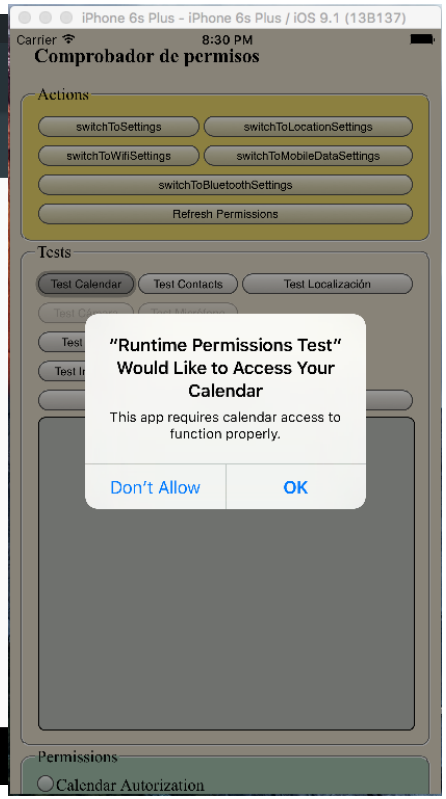


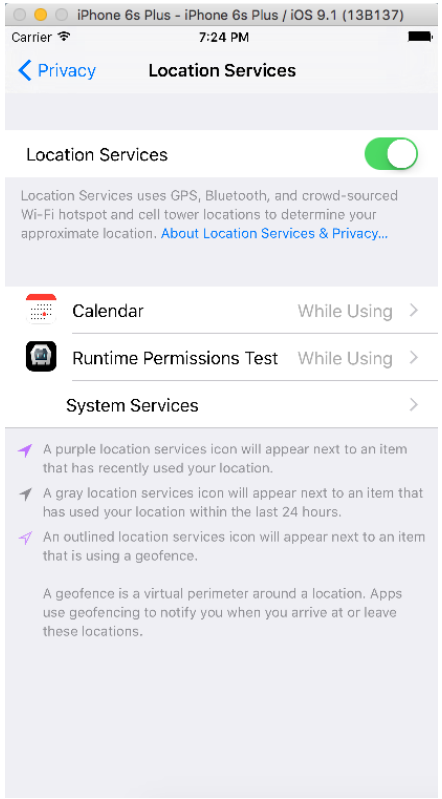


Análisis Comparativo









Análisis Comparativo

Al comparar la gestión de permisos de ambas plataformas, encontramos varias similitudes:

Análisis Comparativo

Al comparar la gestión de permisos de ambas plataformas, encontramos varias similitudes:

- A una aplicación se le otorga permisos básicos al momento de instalación, sin posibilidad de revocarlos.

Análisis Comparativo

Al comparar la gestión de permisos de ambas plataformas, encontramos varias similitudes:

- A una aplicación se le otorga permisos básicos al momento de instalación, sin posibilidad de revocarlos.
- Si una aplicación necesita un permiso no básico, debe requerirlo. El usuario puede otorgarlo o revocarlo.

Análisis Comparativo

Al comparar la gestión de permisos de ambas plataformas, encontramos varias similitudes:

- A una aplicación se le otorga permisos básicos al momento de instalación, sin posibilidad de revocarlos.
- Si una aplicación necesita un permiso no básico, debe requerirlo. El usuario puede otorgarlo o revocarlo.
- Desde la configuración de privacidad, el usuario puede revocar u otorgar permisos a las aplicaciones.

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

- En Android están orientados según el riesgo implícito al otorgarlos.

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

- En Android están orientados según el riesgo implícito al otorgarlos.
- En iOS los permisos están orientados a los componentes.

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

- En Android están orientados según el riesgo implícito al otorgarlos.
- En iOS los permisos están orientados a los componentes.

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

- En Android están orientados según el riesgo implícito al otorgarlos.
- En iOS los permisos están orientados a los componentes.

Sin embargo, se pueden comparar según los componentes que son afectados por un permiso,

Análisis Comparativo

Respecto a cómo se definen los permisos, se observan diferencias de concepto:

- En Android están orientados según el riesgo implícito al otorgarlos.
- En iOS los permisos están orientados a los componentes.

Sin embargo, se pueden comparar según los componentes que son afectados por un permiso, resultando la siguiente Tabla:

Permisos		
Ambas plataformas	*Solo en Android*	*Solo en iOS*
Calendario	-	-
Contactos	-	-
Cámara	-	-
Localización	-	-
-	-	Compartir por Bluetooth
Micrófono	-	-
-	Teléfono	-
Sensores	-	-
-	SMS	-
-	Almacenamiento	-
-	-	*Homekit*
-	-	Redes Sociales
-	-	Diagnóstico
-	-	Publicidad

Análisis Comparativo

Respecto del alcance del sistema de permisos, se observó una falta de granularidad de los permisos que se pueden modificar *en tiempo de ejecución*:

Análisis Comparativo

Respecto del alcance del sistema de permisos, se observó una falta de granularidad de los permisos que se pueden modificar *en tiempo de ejecución*:

- En Android, un permiso es a nivel de grupo. Por lo tanto, el usuario otorga o deniega para todo el grupo.

Análisis Comparativo

Respecto del alcance del sistema de permisos, se observó una falta de granularidad de los permisos que se pueden modificar *en tiempo de ejecución*:

- En Android, un permiso es a nivel de grupo. Por lo tanto, el usuario otorga o deniega para todo el grupo.
- La misma situación ocurre en iOS: se otorga un permiso de acceso a todas las funcionalidades de un determinado componente.

Análisis Comparativo

Respecto del alcance del sistema de permisos, se observó una falta de granularidad de los permisos que se pueden modificar *en tiempo de ejecución*:

- En Android, un permiso es a nivel de grupo. Por lo tanto, el usuario otorga o deniega para todo el grupo.
- La misma situación ocurre en iOS: se otorga un permiso de acceso a todas las funcionalidades de un determinado componente.

Análisis Comparativo

Respecto del alcance del sistema de permisos, se observó una falta de granularidad de los permisos que se pueden modificar *en tiempo de ejecución*:

- En Android, un permiso es a nivel de grupo. Por lo tanto, el usuario otorga o deniega para todo el grupo.
- La misma situación ocurre en iOS: se otorga un permiso de acceso a todas las funcionalidades de un determinado componente.

Como consecuencia de ello, el usuario está delegando a una aplicación demasiados permisos y no tiene expresividad para decir qué funcionalidades autoriza.

Análisis Comparativo

Otra cosa a destacar es la cobertura del sistema de permisos.

Análisis Comparativo

Otra cosa a destacar es la cobertura del sistema de permisos. Las dos plataformas dejan funcionalidades principales del dispositivo sin permisos modificables *en tiempo de ejecución*:

Análisis Comparativo

Otra cosa a destacar es la cobertura del sistema de permisos. Las dos plataformas dejan funcionalidades principales del dispositivo sin permisos modificables *en tiempo de ejecución*:

- En Android se destacan: Acceso a Internet, Compartir vía Bluetooth e Información del Dispositivo.

Análisis Comparativo

Otra cosa a destacar es la cobertura del sistema de permisos. Las dos plataformas dejan funcionalidades principales del dispositivo sin permisos modificables *en tiempo de ejecución*:

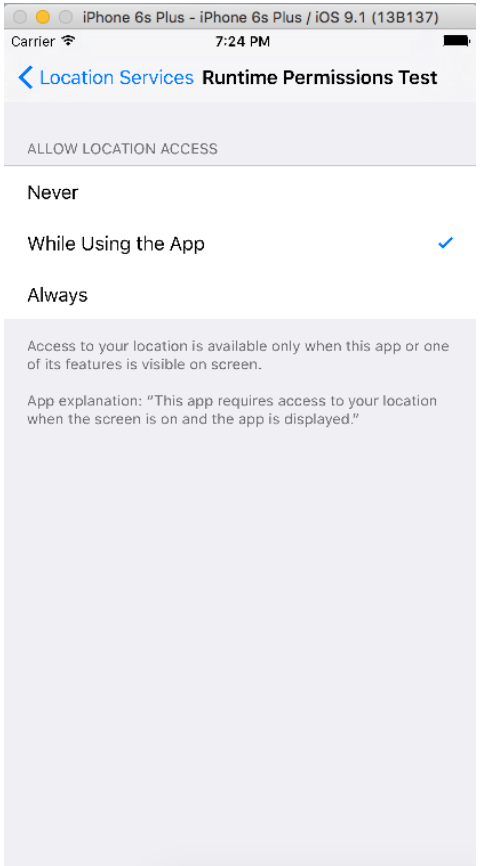
- En Android se destacan: Acceso a Internet, Compartir vía Bluetooth e Información del Dispositivo.
- En iOS se destacan: Acceso a Internet y SMS. Tampoco tiene la suficiente granularidad para administrar el acceso a los datos de las llamadas telefónicas.

Análisis Comparativo

Para finalizar analizamos cómo se muestran al usuario los permisos adquiridos por una aplicación:

Análisis Comparativo

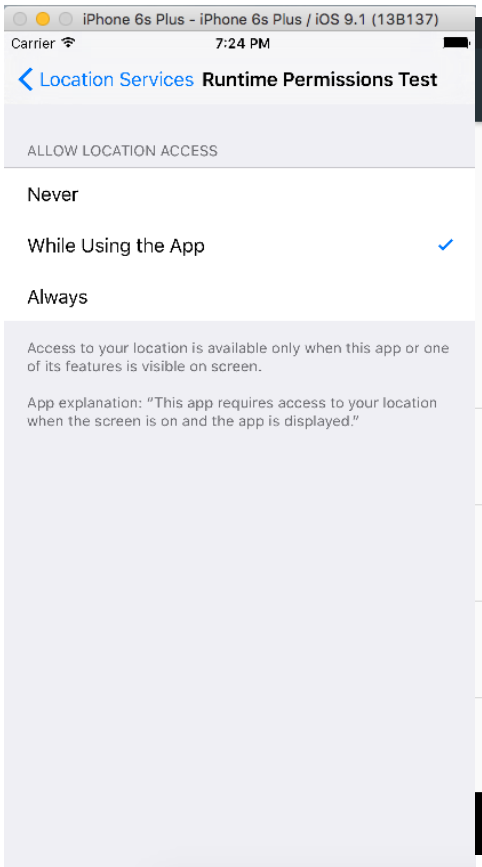
Para finalizar analizamos cómo se muestran al usuario los permisos adquiridos por una aplicación:



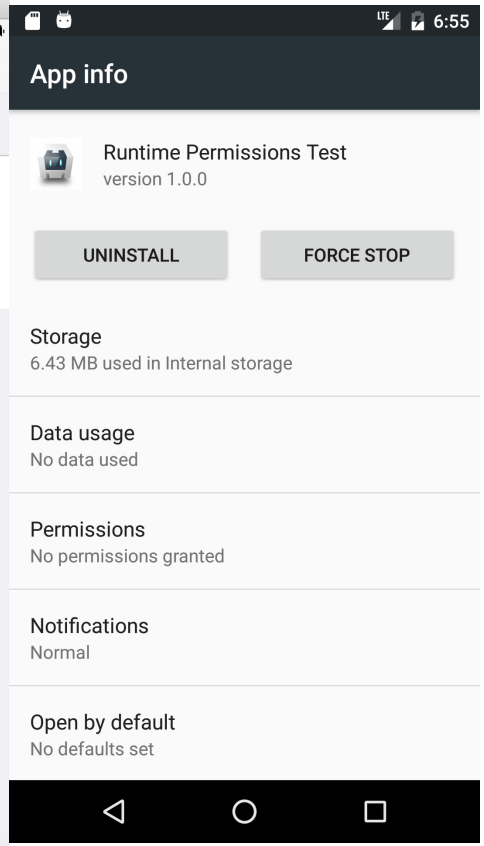
(a) Permisos requeridos por una aplicación.

Análisis Comparativo

Para finalizar analizamos cómo se muestran al usuario los permisos adquiridos por una aplicación:



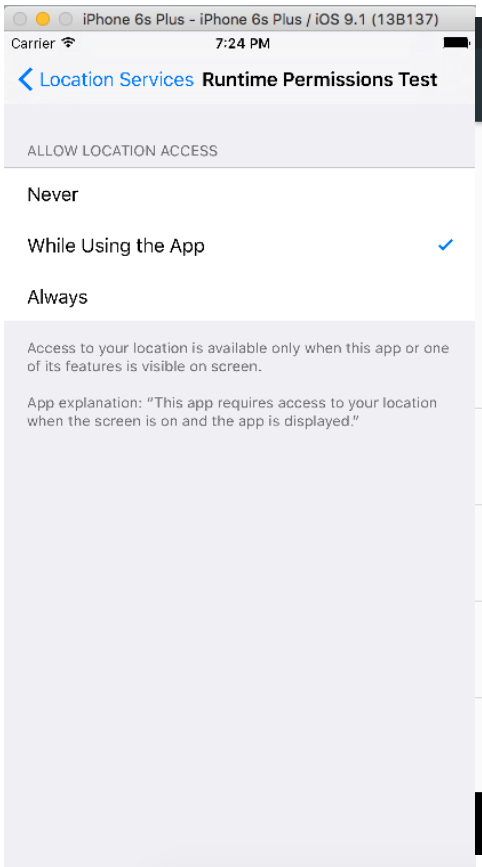
(a) Permisos requeridos por una aplicación.



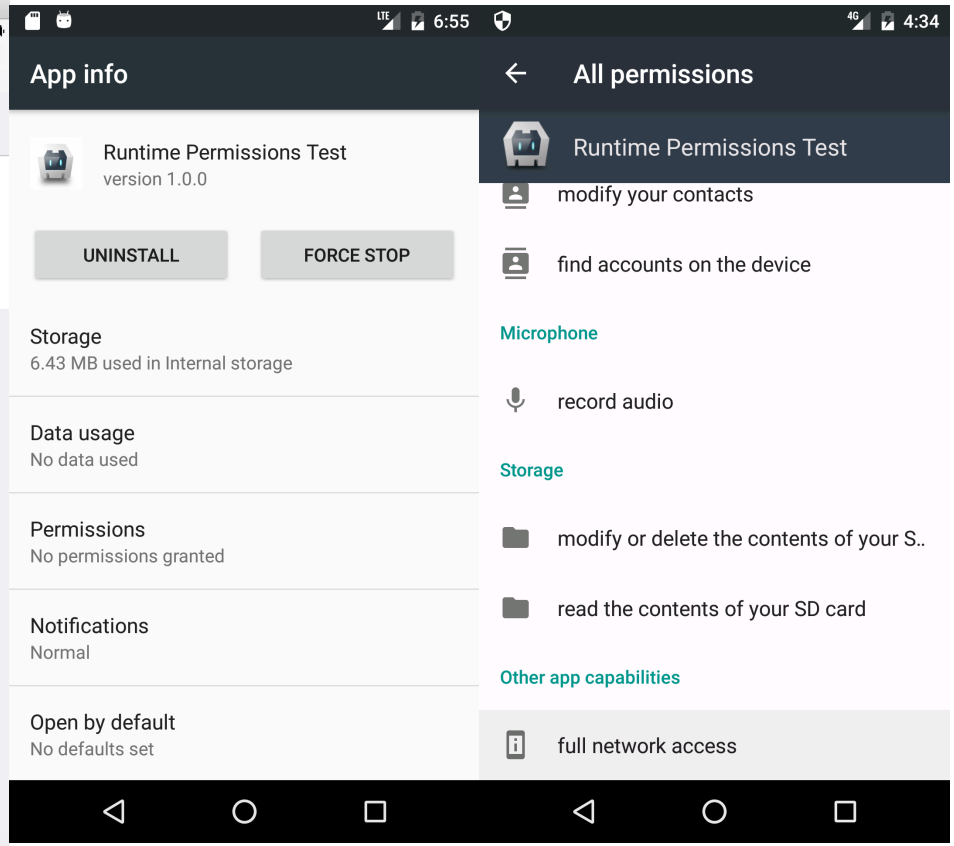
(b) La aplicación no tiene ningún permiso.

Análisis Comparativo

Para finalizar analizamos cómo se muestran al usuario los permisos adquiridos por una aplicación:



(a) Permisos requeridos por una aplicación.



(b) La aplicación no tiene ningún permiso.

(c) Sin embargo, tiene todos los permisos *normales*.

Framework
para la
Comparación de Permisos

Hacia un Framework para la Comparación de Permisos

- Android e iOS permiten cambiar ciertos permisos de una aplicación luego de haberla instalado en el dispositivo.
- Es por ello que se ha desarrollado un *framework* para determinar empíricamente el alcance de los sistemas de permisos de ambas plataformas.

Objetivo I

Se busca dejar en evidencia posibles vulnerabilidades presentes en los modelos de seguridad.

Objetivo II

Se pone énfasis en la relación existente entre la privacidad del usuario y el sistema de permisos, analizando cuál es la cobertura del sistema respecto de los datos sensibles para la privacidad.

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Apache Cordova

Es un *framework* que permite crear aplicaciones para dispositivos móviles utilizado HTML5, CSS3, y JavaScript, con el objetivo de lograr un desarrollo multiplataforma.

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Apache Cordova

Es un *framework* que permite crear aplicaciones para dispositivos móviles utilizado HTML5, CSS3, y JavaScript, con el objetivo de lograr un desarrollo multiplataforma.

El *framework* es una aplicación móvil híbrida

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Apache Cordova

Es un *framework* que permite crear aplicaciones para dispositivos móviles utilizado HTML5, CSS3, y JavaScript, con el objetivo de lograr un desarrollo multiplataforma.

El *framework* es una aplicación móvil híbrida desarrollada con Apache Cordova

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Apache Cordova

Es un *framework* que permite crear aplicaciones para dispositivos móviles utilizado HTML5, CSS3, y JavaScript, con el objetivo de lograr un desarrollo multiplataforma.

El *framework* es una aplicación móvil híbrida desarrollada con Apache Cordova y está compuesto por varios tests.

Hacia un Framework para la Comparación de Permisos

Aplicación Híbrida

Es una aplicación móvil diseñada en un lenguaje de programación web ya sea HTML5, CSS o JavaScript, junto con un *framework* que permite adaptar la vista web a cualquier vista de un dispositivo móvil.

Apache Cordova

Es un *framework* que permite crear aplicaciones para dispositivos móviles utilizado HTML5, CSS3, y JavaScript, con el objetivo de lograr un desarrollo multiplataforma.

El *framework* es una aplicación móvil híbrida desarrollada con Apache Cordova y está compuesto por varios tests.

Cada test pone a prueba a un componente del dispositivo, permitiendo así conocer el alcance de los permisos correspondientes a dicho componente.

Hacia un Framework para la Comparación de Permisos

- Para el presente trabajo se decidió utilizar los emuladores oficiales para testear el *framework* propuesto.

Hacia un Framework para la Comparación de Permisos

- Para el presente trabajo se decidió utilizar los emuladores oficiales para testear el *framework* propuesto.
- Los emuladores permiten interactuar de la misma manera que se haría con un dispositivo real, pero con el ratón y el teclado, y mediante los botones y los controles del emulador.

Hacia un Framework para la Comparación de Permisos

Figura : Áreas del *framework*.

Hacia un Framework para la Comparación de Permisos

Utilizando *framework* se pueden testear:

- Contactos
- Calendario
- Geolocalización
- SMS*
- Sensores
- Almacenamiento
- Información del dispositivo
- Acceso a Internet

Hacia un Framework para la Comparación de Permisos

Utilizando *framework* se pueden testear:

- Contactos
- Calendario
- Geolocalización
- SMS*
- Sensores
- Almacenamiento
- Información del dispositivo
- Acceso a Internet

Sin embargo, no se pueden testear:

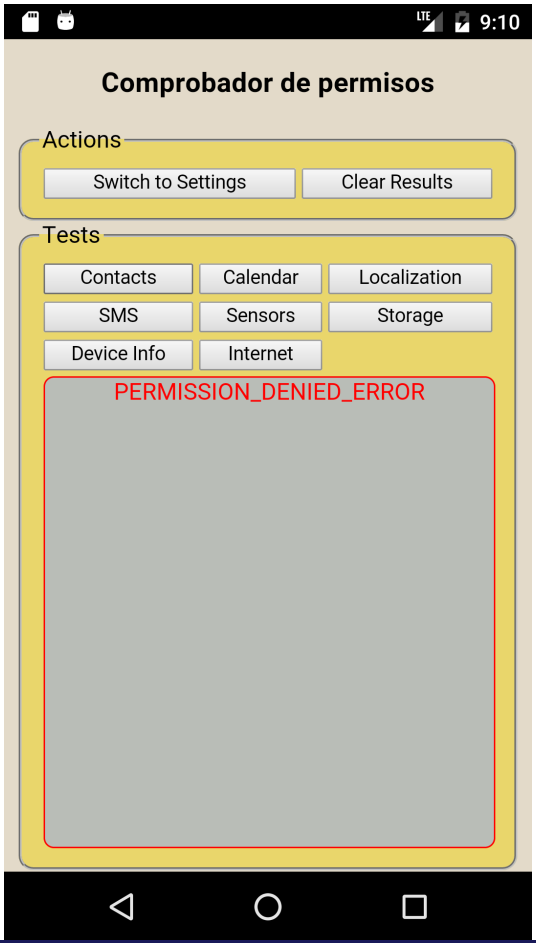
- WIFI
- Bluetooth
- NFC
- Conexiones USB
- Micrófono
- Cámara

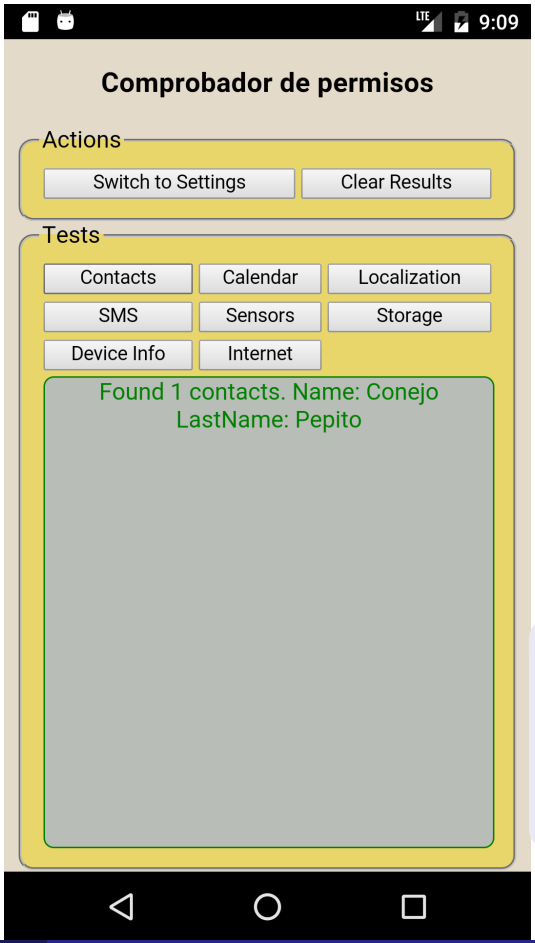
Hacia un Framework para la Comparación de Permisos

Para cada uno de los tests se detalla el algoritmo, los plugins de Apache Cordova que se utilizaron para desarrollarlo y una serie de capturas que muestran los casos exitosos y fallidos.

Hacia un Framework para la Comparación de Permisos

Para cada uno de los tests se detalla el algoritmo, los plugins de Apache Cordova que se utilizaron para desarrollarlo y una serie de capturas que muestran los casos exitosos y fallidos.
Por ejemplo:





Test de Contactos

1: Se imprimen por consola todos los contactos.
2: Se crea un nuevo contacto.
3: Se vuelven a imprimir por consola todos los contactos.

Se utilizó el *plugin* cordova-plugin-contacts (v. 2.3.1) de Apache Cordova.
Para correr el test, es necesario tener el permiso `Contacto`, tanto para Android como para iOS.

Luego de correr los tests, se puenden clasificar los componentes testeados en cuatro clases mutuamente excluyentes, según requieran autorización del usuario para utilizarlos.

Luego de correr los tests, se puenden clasificar los componentes testeados en cuatro clases mutuamente excluyentes, según requieran autorización del usuario para utilizarlos. Dichas clases son:

- Clase A: componentes que requieren autorización explícita en ambas plataformas para poder utilizar las funcionalidades que proveen;
- Clase B: componentes que requieren autorización explícita solamente en Android;
- Clase C: componentes que requieren autorización explícita solamente en iOS;
- Clase D: componentes que no requieren autorización explícita para poder utilizar las funcionalidades que proveen.

Permisos			
Clase A	*Clase B*	*Clase C*	*Clase D*
Contactos	-	-	-
Calendario	-	-	-
Geolocalización	-	-	-
-	SMS	-	-
-	Almacenamiento	-	-
-	-	-	Sensores
-	-	-	Información del dispositivo
-	-	-	Acceso a Internet

Conclusiones
y
Trabajos Futuros

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo,

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos,

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.
Utilizando la medida propuesta, todos los permisos que un usuario puede cambiar en tiempo de ejecución quedan clasificados en tres grupos:

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.
Utilizando la medida propuesta, todos los permisos que un usuario puede cambiar en tiempo de ejecución quedan clasificados en tres grupos: *Ambas Plataformas*,

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.
Utilizando la medida propuesta, todos los permisos que un usuario puede cambiar en tiempo de ejecución quedan clasificados en tres grupos: *Ambas Plataformas*, *Solo en Android*

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.
Utilizando la medida propuesta, todos los permisos que un usuario puede cambiar en tiempo de ejecución quedan clasificados en tres grupos: *Ambas Plataformas*, *Solo en Android* o *Solo en iOS*.

Conclusiones

Primer Aporte

Se realizó un análisis comparativo entre algunas características presentes en los modelos de seguridad de ambas plataformas.
Ellas son: verificación del arranque del sistema operativo, cifrado del sistema de archivos, bloqueo del dispositivo y sistemas de permisos.
Todas ellas se eligieron porque son importantes a la hora de resguardar la privacidad del usuario.

Segundo Aporte

Como fruto del análisis, se logró establecer una medida de comparación entre los permisos presentes en ambas plataformas. La medida propuesta es la siguiente: *dos permisos son similares si resguardan un componente que provee la misma funcionalidad*.
Utilizando la medida propuesta, todos los permisos que un usuario puede cambiar en tiempo de ejecución quedan clasificados en tres grupos: *Ambas Plataformas*, *Solo en Android* o *Solo en iOS*.
Cabe aclarar que los grupos son mutuamente excluyentes.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales:

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos;

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*.

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*. Utilizando el criterio propuesto, se obtuvieron cuatro clases mutuamente excluyentes:

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*. Utilizando el criterio propuesto, se obtuvieron cuatro clases mutuamente excluyentes:

- Clase A: componentes que requieren autorización explícita en ambas plataformas para poder utilizar las funcionalidades que proveen;

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*. Utilizando el criterio propuesto, se obtuvieron cuatro clases mutuamente excluyentes:

- Clase A: componentes que requieren autorización explícita en ambas plataformas para poder utilizar las funcionalidades que proveen;
- Clase B: componentes que requieren autorización explícita solamente en Android;

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*. Utilizando el criterio propuesto, se obtuvieron cuatro clases mutuamente excluyentes:

- Clase A: componentes que requieren autorización explícita en ambas plataformas para poder utilizar las funcionalidades que proveen;
- Clase B: componentes que requieren autorización explícita solamente en Android;
- Clase C: componentes que requieren autorización explícita solamente en iOS;

Conclusiones

Tercer Aporte

Otro aporte es el *Framework para la Comparación de Permisos*. Tiene dos funciones principales: determinar empíricamente los alcances de los sistemas de permisos; y establecer una relación entre los permisos presenten en las dos plataformas.
El *framework* está compuesto por una batería de tests, teniendo cada uno de ellos la tarea de probar una funcionalidad provista por Android e iOS.

Cuarto Aporte

Como resultado de la utilización del *framework* se determinó una clasificación de permisos. El criterio utilizado fue: *un componente pertenece a una clase según requiera autorización explícita del usuario para utilizarlo*. Utilizando el criterio propuesto, se obtuvieron cuatro clases mutuamente excluyentes:

- Clase A: componentes que requieren autorización explícita en ambas plataformas para poder utilizar las funcionalidades que proveen;
- Clase B: componentes que requieren autorización explícita solamente en Android;
- Clase C: componentes que requieren autorización explícita solamente en iOS;
- Clase D: componentes que no requieren autorización explícita para poder utilizar las funcionalidades que proveen.

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.
- Desarrollar tests para las funcionalidades que no pueden ser emuladas (ver Seccion 5.1.1).

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.
- Desarrollar tests para las funcionalidades que no pueden ser emuladas (ver Seccion 5.1.1).
- Desarrollar un test para poder comparar el cifrado de archivos.

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.
- Desarrollar tests para las funcionalidades que no pueden ser emuladas (ver Seccion 5.1.1).
- Desarrollar un test para poder comparar el cifrado de archivos.
- Android otorga todos los permisos *normales*, tal como se enuncia en la Seccion 3.1.4. Pero no sabemos qué permisos otorga iOS. Se podrían desarrollar varios test para descubrirlos.

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.
- Desarrollar tests para las funcionalidades que no pueden ser emuladas (ver Seccion 5.1.1).
- Desarrollar un test para poder comparar el cifrado de archivos.
- Android otorga todos los permisos *normales*, tal como se enuncia en la Seccion 3.1.4. Pero no sabemos qué permisos otorga iOS. Se podrían desarrollar varios test para descubrirlos.
- Dado que salieron al mercado las versiones Android 8.0 e iOS 11, se podría analizar extender el *framework* para las características de seguridad adicionadas en dichas versiones.

Trabajos futuros

Para finalizar, se enumeran algunas líneas a seguir como trabajos a futuro:

- Probar y mejorar los test que actualmente conforman el framework en dispositivos reales.
- Desarrollar tests para las funcionalidades que no pueden ser emuladas (ver Seccion 5.1.1).
- Desarrollar un test para poder comparar el cifrado de archivos.
- Android otorga todos los permisos *normales*, tal como se enuncia en la Seccion 3.1.4. Pero no sabemos qué permisos otorga iOS. Se podrían desarrollar varios test para descubrirlos.
- Dado que salieron al mercado las versiones Android 8.0 e iOS 11, se podría analizar extender el *framework* para las características de seguridad adicionadas en dichas versiones.
- En el Capítulo 4 se mencionan algunos análisis previos relacionados a la seguridad de Android y/o de iOS. Se podría profundizar más en comparar el modelo propuesto en el presente informe con los análisis mencionados previamente.

¿ Preguntas ?

¡¡ Mil gracias por venir !!